UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/689,126	10/20/2003	John B. Condon	BLD920030025US1	2905

50441 7590 04/24/2009
DUFT BORNSEN & FISHMAN, LLP
1526 SPRUCE STREET
SUITE 302
BOULDER, CO 80302

EXAMINER	
DICKERSON, CHAD S	

ART UNIT	PAPER NUMBER
2625	

MAIL DATE	DELIVERY MODE
04/24/2009	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	**Application No.** 10/689,126	**Applicant(s)** CONDON ET AL.
	Examiner CHAD DICKERSON	**Art Unit** 2625

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *17 February 2009*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-4,9-13,18-21 and 25* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-4, 9-13, 18-21 and 25* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *10/20/2003* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.

4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .

5) ☐ Notice of Informal Patent Application

6) ☐ Other: _____.

DETAILED ACTION

Response to Arguments

1. Applicant's arguments, see page 8, filed 2/17/2009, with respect to the 101 rejections have been fully considered and are persuasive. The 101 rejections of claims 1-4 and 9 have been withdrawn.

2. Applicant's arguments, see pages 8-13, filed 2/17/2009, with respect to the rejection(s) of claim(s) 1-4, 9-13, 18-21 and 25 under 103(a) have been fully considered and are persuasive. Therefore, the rejection has been withdrawn. However, upon further consideration, a new ground(s) of rejection is made in view of the reference of Vennekens '711 (USP 5652711). When reading the Vennekens reference that was cited by the previously used reference of Wood '934, the Examiner realized that this reference performs the features of the claim limitations. In particular, the reference discloses processing different segments of PDLs, in an independent fashion that does not require certain data segments processed before other segments. The Vennekens '711 reference is used to replace the Wood reference while still applying the McIntyre and Salgado references. However, even with the introduction of the new reference, the Examiner would like to briefly address an allegation regarding the use of the Salgado reference in the previous Office action.

In Applicant's remarks, the Applicant stated that the Examiner used improper hindsight knowledge to piece together a rejection to render the claimed invention obvious. However, the Examiner respectfully disagrees with this assertion because the Salgado reference does divide a job into units, or blocks, that are transferred around the image processing system for output. For example, in figure 4 and shown in column 10, the Salgado discloses images of a job are stored in memory as a series of blocks that are transferred in a series of packets. These divisions in data when being transferred can be considered as a job being parsed into a plurality of work units that are in a first format[1]. With that being said, the Examiner believes that this is enough reason to combine the references of McIntyre, Vennekens and Salgado, since the data in Salgado is divided in a specific manner in order to transfer the data for further processing. Therefore, with the above reasoning regarding the Salgado reference, the Examiner believes it is still appropriate to combine the references currently applied.

Claim Rejections - 35 USC § 103

3. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

4. Claims 1-4, 9-13, 18-21 and 25 are rejected under 35 U.S.C. 103(a) as being unpatentable over McIntyre '478 (USP 6690478) in view of Vennekens '711 (USP 5652711) and Salgado '621 (USP 6504621).

Re claim 1: McIntyre '478 discloses a method and apparatus for utilizing multiple versions of a page descriptor language comprising the steps of:

[1] See Salgado '621 at col. 9, ln 56 – col. 10, ln 44.

processing each of the plurality of work units by at least one compute node to convert each work unit into a second format (i.e. in McIntyre '478, a plurality of print jobs, considered as work units can be processed. The printer driver (114), considered as the compute node, processes the incoming print jobs by recognizing the type of PDL is input into the system. The printer driver selects a PDL type from a PDL registry (112) to correspond with the incoming data and this PDL type chosen is used to convert the data into a different format, analogous to the second format, which is a low-level data stream; see figs. 1 and 3; col. 2, lines 18-30, col. 4, lines 11-66, col. 5, lines 160 and col. 6, lines 1-25).

However, McIntyre '478 fails to teach parsing the datastream into a plurality of work units in a first format wherein each work unit may be processed independent of all other work units, wherein the plurality of work units are parsed from a single job, wherein each work unit may either be a data work unit or a control work unit, queuing each data work unit on a queue accessible by a plurality of compute nodes, wherein the processing of each work unit is independent of processing of the other work units and wherein multiple work units are processed in parallel by multiple compute nodes.

However, this is well known in the art as evidenced by Vennekens '711. Vennekens '711 discloses parsing the datastream into a plurality of work units in a first format (i.e. the inventions of McIntyre and Vennekens are both involved with processing PDL for an output in the system (same filed of endeavor). However, the apparatus and method of Vennekens '711 divides the PDL data stream to provide a plurality of PDL segments. These segments are in a PDL format, analogous to a first format and the function of parsing is performed by the generation of the PDL data stream into a plurality of independent PDL segments, which the plurality of segments are considered as the plurality of work units; see fig. 1, col. 2, line 55 - col. 3, line 65 and col. 4, lines 24-39)

wherein each work unit may be processed independent of all other work units (i.e. the master process (32) is used to analyze the incoming PDL and generate independent PDL segments that can be processed independently from other segments; see col. 6, lines 19-62),

wherein the plurality of work units are parsed from a single job (i.e. a page that can represent a single job can be described by a plurality of independent PDL data segments; see col. 3, line 66 – col. 4, line 23),

wherein each work unit may either be a data work unit or a control work unit (i.e. when the PDL data stream arrives at the master process (32) the data can be subdivided into data commands and control commands that are considered analogous to the data and control work units; see col. 4, line 46 - col. 5, line 14);

queuing each data work unit on a queue accessible by a plurality of compute nodes (i.e. when the PDL data stream is divided into the data or control commands, then the commands are then passed from the master process to a FIFO queue, which is considered as the mechanism that queues work units; see fig. 1, col. 6, lines 19-65) and

a control work unit processed by a computer node (i.e. in the system, a PDL data stream segment is assigned a control command and the segment with the control commands is translated with a sub-process (36 or 37), which is considered as a computer node; see col. 2, line 55 - col. 3, line 65),

wherein the processing of each work unit is independent of processing of the other work units (i.e. the master process (32) is used to analyze the incoming PDL and generate independent PDL segments that can be processed independently from other segments. The work units are then processed independently in the sub-processes (36 or 37) in the system; see col. 6, lines 19-62) and

wherein multiple work units are processed in parallel by multiple compute nodes (i.e. the PDL data stream is divided into segments that are processed in parallel on the sub-processes; see col. 2, line 55 - col. 3, line 65).

Therefore, in view of Vennekens '711, it would have been obvious to one of ordinary skill at the time the invention was made to parsing the datastream into a plurality of work units in a first format wherein each work unit may be processed independent of all other work units, wherein the plurality of work units are parsed from a single job, wherein each work unit may either be a data work unit or a control work unit, queuing each data work unit on a queue accessible by a plurality of compute nodes, wherein the processing of each work unit is independent of processing of the other work units and wherein multiple work units are processed in parallel by multiple compute nodes, incorporated in the device of McIntyre, in order to generate a plurality of independent PDL data stream segments (as stated in Vennekens '711 col. 2, lines 58-65).

However, the combination of McIntyre ‘478 and Vennekens '711 and fails to teach wherein each control unit may be an immediate control work unit or a scheduled work unit or an interrupt control work unit; queuing a scheduled control work unit at a tail of the queue to be processed by a compute node after all other work units presently in the queue; queuing an immediate control work unit at a head of the queue to be processed by a compute node before all other work units in the queue; forwarding an interrupt control work unit to a compute node immediately regardless of any work units in the queue.

However, this is well known in the art as evidenced by Salgado ‘621. Salgado ‘621 discloses wherein each control unit may be an immediate control work unit or a scheduled work unit or an interrupt control work unit (i.e. like to previously applied references, the reference of Salgado is used to process PDL information for output (same field of endeavor). Also, the Salgado reference discloses the image data being divided into blocks, similar to the division of PDL information in Vennekens '711. However, with the different types of jobs in PDL being queued in Example 1, each type of job reflects a different type of control work unit. For example, the interrupt control work unit is analogous to the Authorized User Job in EXAMPLE 1 in column 19. Once this AU job reaches the system, it is immediately processed and interrupts the job that is currently being processed. The net print job is considered analogous to the immediate control work unit since this job is immediately placed at the head of the queue before all work units in the queue. This job is also similar to the copy job listed in the same column 18. The system report error job is considered as the schedule control work unit since it is paced at the end of the queue after all the other jobs presently in the queue; see col. 18, line 40 – col. 20 36);

queueing a scheduled control work unit at a tail of the queue to be processed after all other work units presently in the queue (i.e. when looking at EXAMPLE 1 in column 18, this example shows examples of the different types of control work units. The system error report job is considered as the schedule work unit, placed at the end of the queue and not processed until the jobs before the system error report job are processed beforehand; see col. 18, line 40 – col. 20, line 36);

queueing an immediate control work unit at a head of the queue to be processed before all other work units in the queue (i.e. the net print job is considered analogous to the immediate control work unit since this job is immediately placed at the head of the queue before all work units in the queue. This job is also similar to the copy job listed in the same column 18; see col. 18, line 40 – col. 20, line 36);

forwarding an interrupt control work unit immediately regardless of any work units in the queue (i.e. the interrupt control work unit is analogous to the Authorized User Job in EXAMPLE 1 in column 19. Once this AU job reaches the system, it is immediately processed and interrupts the job that is currently being processed; see col. 18, line 40 – col. 20, line 36).

Therefore, in view of Salgado '621, it would have been obvious to one of ordinary skill at the time the invention was made to have the feature of wherein each control unit may be an immediate control work unit or a scheduled work unit or an interrupt control work unit; queuing a scheduled control work unit at a tail of the queue to be processed after all other work units presently in the queue; queuing an immediate control work unit at a head of the queue to be processed before all other work units in the queue; forwarding an interrupt control work unit immediately regardless of any work units in the queue incorporated in the device of McIntyre '478, as combined with the features of Vennekens '711, in order to have a system with a queue for structuring an order in which a plurality of print jobs is to be processed (as stated in Salgado col. 5, lines 6-29).

Re claim 2: The teachings of McIntyre '478 in view of Vennekens '711 and Salgado '621 are disclosed above.

McIntyre '478 discloses the method, wherein the parsing step (a) includes:

providing a plurality of sources, wherein each source is associated with at least one transform (i.e. in McIntyre '478, a method for registration and selection of multiple page description languages (i.e. personalities) is presented. The personalities, analogous to a plurality of sources, are associated with a transform, or conversion, in order to convert the input instructions into a printer dependent data stream interpreted by the printing subsystem (108) to produce an output page; see figs. 1 and 3; col. 2, lines 18-30, col. 4, lines 11-66, col. 5, lines 160 and col. 6, lines 1-25);

instantiating at least one source of the plurality of sources, wherein the at least one instantiated source is associated with the datastream format (i.e. the printer driver (114) recognizes a realization of the personality related to the transform, or instantiates one transform, in order to perform a conversion, that is associated with received instructions. The printer driver (114) analyzes these received instructions and chooses one of the multiple PDLs registered in the PDL registry (112); see figs. 1 and 3; col. 2, lines 18-30, col. 4, lines 11-66, col. 5, lines 160 and col. 6, lines 1-25); and

utilizing the at least one source (i.e. once the printer driver (114) finds an appropriate version of a PDL registered within the PDL registry (112), the printer driver invokes the personality, or utilizes the personality analogous to the source, to convert the instructions from a high level language to a printer dependent data stream; see figs. 1 and 3; col. 2, lines 18-30, col. 4, lines 11-66, col. 5, lines 160 and col. 6, lines 1-25).

However, McIntyre '478 fails to teach to parse the datastream.

However, this is well known in the art as evidenced by Vennekens '711. Vennekens '711 discloses to parse the datastream (i.e. the inventions of McIntyre and Vennekens are both involved with processing PDL for an output in the system (same filed of endeavor). However, the apparatus and method of Vennekens '711 divides the PDL data stream to provide a plurality of PDL segments. These segments are in a PDL format, analogous to a first format and the function of parsing is performed by the generation of the PDL data stream into a plurality of independent PDL segments, which the plurality of segments are considered as the plurality of work units; see fig. 1, col. 2, line 55 - col. 3, line 65 and col. 4, lines 24-39).

Therefore, in view of Vennekens '711, it would have been obvious to one of ordinary skill at the time the invention was made to parse the datastream in order to generate a plurality of independent PDL data stream segments (as stated in Vennekens '711 col. 2, lines 58-65).

Re claim 3: The teachings of McIntyre '478 in view of Vennekens '711 and Salgado ‘621 are disclosed above.

McIntyre '478 discloses the method, wherein the processing step (b) includes:

loading the at least one transform associated with the at least one instantiated source in the at least one compute node (i.e. the printer driver (114) recognizes a transform, or conversion, associated with at least one realization of the personality, analogous to an instantiated source, that is able to perform the transform and invokes the personality, or PDL, to perform the conversion of the received instructions. The action of invoking the personality after associated the PDL with the received data is analogous to immediately loading the transform in order to be utilized for transformation of the received data; see figs. 1 and 3; col. 2, lines 18-30, col. 4, lines 11-66, col. 5, lines 160 and col. 6, lines 1-25); and

utilizing the at least one transform to convert a work unit of the plurality of work units from the first format to the second format (i.e. in the system of McIntyre '478, the system is able to process a plurality of print jobs, considered as work units. The PDL utilized by the printer driver (114) to convert the received instructions, or print job, from a high level language to a printer dependent data, or language, is analogous to converting a work unit from a first format to a second format; see figs. 1 and 3; col. 2, lines 18-30, col. 4, lines 11-66, col. 5, lines 160 and col. 6, lines 1-25).

Re claim 4: The teachings of McIntyre '478 in view of Wood '934 and Salgado ‘621 are disclosed above.

However, McIntyre '478 fails to teach the method further comprising: (c) load balancing the plurality of work units.

However, this is well known in the art as evidenced by Vennekens '711. Vennekens '711 discloses the method further comprising: (c) load balancing the plurality of work units (i.e. the inventions of McIntyre and Vennekens are both involved with processing PDL for an output in the system (same filed of endeavor). However, the apparatus and method of Vennekens '711 divides the PDL data stream to provide a plurality of PDL segments. The function of load balancing among the sub-process is performed; see col. 6, lines 19-62).

Therefore, in view of Vennekens '711, it would have been obvious to one of ordinary skill at the time the invention was made to have the step of load balancing the plurality of work units in order to generate a plurality of independent PDL data stream segments (as stated in Vennekens '711 col. 2, lines 58-65).

Re claim 9: The teachings of McIntyre '478 in view of Wood '934 and Salgado '621 are disclosed above.

McIntyre '478 discloses the method, wherein the at least one source is instantiated as a dynamic library (i.e. when using a transform to convert incoming data into another form, the printer driver (114) requests for a transform to convert from the incoming PDL into a low-level language for the printer to understand. This is performed by linking the incoming data to the specific personality that will perform the transformation of the data stream to the low-level format. This process is similar to a library with a collection of subprograms used to develop other pieces of information used by the system and provide the function of transformation that is linked to a certain input language in the invention of McIntyre '478; see figs. 1 and 3; col. 2, lines 18-30, col. 4, lines 11-66, col. 5, lines 160 and col. 6, lines 1-25).

Re claim 10: McIntyre '478 discloses a method and apparatus for utilizing multiple versions of a page descriptor language, the program instructions for:

processing each of the plurality of work units by at least one compute node to convert each data work unit into a second format (i.e. in McIntyre '478, a plurality of print jobs, considered as work units can be processed. The printer driver (114), considered as the compute node, processes the incoming print jobs by recognizing the type of PDL is input into the system. The printer driver selects a PDL type from a PDL registry (112) to correspond with the incoming data and this PDL type chosen is used to convert the data into a different format, analogous to the second format, which is a low-level data stream. Also, with the above actions capable of being performed on a storage medium having stored the executable instructions to implement the teachings of the invention of McIntyre '478, the above feature of a computer readable medium containing program instructions is performed; see figs. 1, 3 and 6; col. 2, lines 18-30, col. 4, lines 11-66, col. 5, lines 160, col. 6, lines 1-25 and col. 7, lines 7-23).

However, McIntyre '478 fails to teach parsing the datastream into a plurality of work units in a first format wherein each work unit may be processed independent of all other work units, wherein the plurality of work units are parsed from a single job, wherein each work unit may either be a data work unit or a control work unit, queueing each data work unit on a queue accessible by a plurality of compute nodes, wherein the processing of each work unit is independent of processing of the other work units and wherein multiple work units are processed in parallel by multiple compute nodes.

However, this is well known in the art as evidenced by Vennekens '711. Vennekens '711 discloses parsing the datastream into a plurality of work units in a first format (i.e. the inventions of McIntyre and Vennekens are both involved with processing PDL for an output in the system (same filed of endeavor). However, the apparatus and method of Vennekens '711 divides the PDL data stream to provide a plurality of PDL segments. These segments are in a PDL format, analogous to a first format and the function of parsing is performed by the generation of the PDL data stream into a plurality of independent PDL segments, which the plurality of segments are considered as the plurality of work units; see fig. 1, col. 2, line 55 - col. 3, line 65 and col. 4, lines 24-39)

wherein each work unit may be processed independent of all other work units (i.e. the master process (32) is used to analyze the incoming PDL and generate independent PDL segments that can be processed independently from other segments; see col. 6, lines 19-62),

wherein the plurality of work units are parsed from a single job (i.e. a page that can represent a single job can be described by a plurality of independent PDL data segments; see col. 3, line 66 – col. 4, line 23), wherein each work unit may either be a data work unit or a control work unit (i.e. when the PDL data stream arrives at the master process (32) the data can be subdivided into data commands and control commands that are considered analogous to the data and control work units; see col. 4, line 46 - col. 5, line 14);

queuing each data work unit on a queue accessible by a plurality of compute nodes (i.e. when the PDL data stream is divided into the data or control commands, then the commands are then passed from the master process to a FIFO queue, which is considered as the mechanism that queues work units; see fig. 1, col. 6, lines 19-65) and

a control work unit processed by a computer node (i.e. in the system, a PDL data stream segment is assigned a control command and the segment with the control commands is translated with a sub-process (36 or 37), which is considered as a computer node; see col. 2, line 55 - col. 3, line 65),

wherein the processing of each work unit is independent of processing of the other work units (i.e. the master process (32) is used to analyze the incoming PDL and generate independent PDL segments that can be processed independently from other segments. The work units are then processed independently in the sub-processes (36 or 37) in the system; see col. 6, lines 19-62) and

wherein multiple work units are processed in parallel by multiple compute nodes (i.e. the PDL data stream is divided into segments that are processed in parallel on the sub-processes; see col. 2, line 55 - col. 3, line 65).

Therefore, in view of Vennekens '711, it would have been obvious to one of ordinary skill at the time the invention was made to parsing the datastream into a plurality of work units in a first format wherein each work unit may be processed independent of all other work units, wherein the plurality of work units are parsed from a single job, wherein each work unit may either be a data work unit or a control work unit, queuing each data work unit on a queue accessible by a plurality of compute nodes, wherein the processing of each work unit is independent of processing of the other work units and wherein multiple work units are processed in parallel by multiple compute nodes, incorporated in the device of McIntyre, in order to generate a plurality of independent PDL data stream segments (as stated in Vennekens '711 col. 2, lines 58-65).

However, the combination of McIntyre '478 and Vennekens '711 fails to teach wherein each control unit may be an immediate control work unit or a scheduled work unit or an interrupt control work unit; queuing a scheduled control work unit at a tail of the queue to be processed after all other work units presently in the queue; queuing an immediate control work unit at a head of the queue to be processed before all other work units in the queue; forwarding an interrupt control work unit immediately regardless of any work units in the queue.

However, this is well known in the art as evidenced by Salgado '621. Salgado '621 discloses wherein each control unit may be an immediate control work unit or a scheduled work unit or an interrupt control work unit (i.e. like the above applied references, the reference of Salgado is used to process PDL information for output (same field of endeavor). However, with the different types of jobs being queued in Example 1, each type of job reflects a different type of control work unit. For example, the interrupt control work unit is analogous to the Authorized User Job in EXAMPLE 1 in column 19. Once this AU job reaches the system, it is immediately processed and interrupts the job that is currently being processed. The net print job is considered analogous to the immediate control work unit since this job is immediately placed at the head of the queue before all work units in the queue. This job is also similar to the copy job listed in the same column 18. The system report error job is considered as the schedule control work unit since it is paced at the end of the queue after all the other jobs presently in the queue; see col. 18, line 40 – col. 20 36);

queueing a scheduled control work unit at a tail of the queue to be processed after all other work units presently in the queue (i.e. when looking at EXAMPLE 1 in column 18, this example shows examples of the different types of control work units. The system error report job is considered as the schedule work unit, placed at the end of the queue and not processed until the jobs before the system error report job are processed beforehand; see col. 18, line 40 – col. 20, line 36);

queueing an immediate control work unit at a head of the queue to be processed before all other work units in the queue (i.e. the net print job is considered analogous to the immediate control work unit since this job is immediately placed at the head of the queue before all work units in the queue. This job is also similar to the copy job listed in the same column 18; see col. 18, line 40 – col. 20, line 36);

forwarding an interrupt control work unit immediately regardless of any work units in the queue (i.e. the interrupt control work unit is analogous to the Authorized User Job in EXAMPLE 1 in column 19. Once this AU job reaches the system, it is immediately processed and interrupts the job that is currently being processed; see col. 18, line 40 – col. 20, line 36).

Therefore, in view of Salgado ‘621, it would have been obvious to one of ordinary skill at the time the invention was made to have the feature of wherein each control unit may be an immediate control work unit or a scheduled work unit or an interrupt control work unit; queuing a scheduled control work unit at a tail of the queue to be processed after all other work units presently in the queue; queuing an immediate control work unit at a head of the queue to be processed before all other work units in the queue; forwarding an interrupt control work unit immediately regardless of any work units in the queue incorporated in the device of McIntyre ‘478, as combined with the features of Vennekens '711, in order to have a system with a queue for structuring an order in which a plurality of print jobs is to be processed (as stated in Salgado col. 5, lines 6-29).

Re claim 11: The teachings of McIntyre '478 in view of Vennekens '711 and Salgado ‘621 are disclosed above.

McIntyre ‘478 discloses the computer readable medium of claim 10, wherein the parsing instruction (a) includes:

providing a plurality of sources, wherein each source is associated with at least one transform (i.e. in McIntyre '478, a method for registration and selection of multiple page description languages (i.e. personalities) is presented. The personalities, analogous to a plurality of sources, are associated with a transform, or conversion, in order to convert the input instructions into a printer dependent data stream interpreted by the printing subsystem (108) to produce an output page; see figs. 1 and 3; col. 2, lines 18-30, col. 4, lines 11-66, col. 5, lines 160 and col. 6, lines 1-25);

instantiating at least one source of the plurality of sources, wherein the at least one instantiated source is associated with the datastream format (i.e. the printer driver (114) recognizes a realization of the personality related to the transform, or instantiates one transform, in order to perform a conversion, that is associated with received instructions. The printer driver (114) analyzes these received instructions and chooses one of the multiple PDLs registered in the PDL registry (112); see figs. 1 and 3; col. 2, lines 18-30, col. 4, lines 11-66, col. 5, lines 160 and col. 6, lines 1-25); and

utilizing the at least one source (i.e. once the printer driver (114) finds an appropriate version of a PDL registered within the PDL registry (112), the printer driver invokes the personality, or utilizes the personality analogous to the source, to convert the instructions from a high level language to a printer dependent data stream; see figs. 1 and 3; col. 2, lines 18-30, col. 4, lines 11-66, col. 5, lines 160 and col. 6, lines 1-25).

However, McIntyre '478 fails to teach to parse the datastream.

However, this is well known in the art as evidenced by Vennekens '711. Vennekens '711 discloses to parse the datastream (i.e. the inventions of McIntyre and Vennekens are both involved with processing PDL for an output in the system (same filed of endeavor). However, the apparatus and method of Vennekens '711 divides the PDL data stream to provide a plurality of PDL segments. These segments are in a PDL format, analogous to a first format and the function of parsing is performed by the generation of the PDL data stream into a plurality of independent PDL segments, which the plurality of segments are considered as the plurality of work units; see fig. 1, col. 2, line 55 - col. 3, line 65 and col. 4, lines 24-39).

Therefore, in view of Vennekens '711, it would have been obvious to one of ordinary skill at the time the invention was made to parse the datastream in order to generate a plurality of independent PDL data stream segments (as stated in Vennekens '711 col. 2, lines 58-65).

Re claim 12: The teachings of McIntyre '478 in view of Vennekens '711 and Salgado ‘621 are disclosed above.

McIntyre ‘478 discloses the computer readable medium of claim 11, wherein the processing instruction includes:

loading the at least one transform associated with the at least one instantiated source in the at least one compute node (i.e. the printer driver (114) recognizes a transform, or conversion, associated with at least one realization of the personality, analogous to an instantiated source, that is able to perform the transform and invokes the personality, or PDL, to perform the conversion of the received instructions. The action of invoking the personality after associated the PDL with the received data is analogous to immediately loading the transform in order to be utilized for transformation of the received data; see figs. 1 and 3; col. 2, lines 18-30, col. 4, lines 11-66, col. 5, lines 160 and col. 6, lines 1-25); and

utilizing the at least one transform to convert a work unit of the plurality of work units from the first format to the second format (i.e. in the system of McIntyre '478, the system is able to process a plurality of print jobs, considered as work units. The PDL utilized by the printer driver (114) to convert the received instructions, or print job, from a high level language to a printer dependent data, or language, is analogous to converting a work unit from a first format to a second format; see figs. 1 and 3; col. 2, lines 18-30, col. 4, lines 11-66, col. 5, lines 160 and col. 6, lines 1-25).

Re claim 13: The teachings of McIntyre '478 in view of Vennekens '711 and Salgado ‘621 are disclosed above.

However, McIntyre '478 fails to teach the computer readable medium further comprising: load balancing the plurality of work units.

However, this is well known in the art as evidenced by Vennekens '711. Vennekens '711 discloses the computer readable medium further comprising: (c) load balancing the plurality of work units (i.e. the inventions of McIntyre and Vennekens are both involved with processing PDL for an output in the system (same filed of endeavor). However, the apparatus and method of Vennekens '711 divides the PDL data stream to provide a plurality of PDL segments. The function of load balancing among the sub-process is performed; see col. 6, lines 19-62).

Therefore, in view of Vennekens '711, it would have been obvious to one of ordinary skill at the time the invention was made to have the step of load balancing the plurality of work units in order to generate a plurality of independent PDL data stream segments (as stated in Vennekens '711 col. 2, lines 58-65).

Re claim 18: The teachings of McIntyre '478 in view of Vennekens '711 and Salgado ‘621 are disclosed above.

McIntyre '478 discloses the computer readable medium, wherein the at least one source is instantiated as a dynamic library (i.e. when using a transform to convert incoming data into another form, the printer driver (114) requests for a transform to convert from the incoming PDL into a low-level language for the printer to understand. This is performed by linking the incoming data to the specific personality that will perform the transformation of the data stream to the low-level format. This process is similar to a library with a collection of subprograms used to develop other pieces of information used by the system and provide the function of transformation that is linked to a certain input language in the invention of McIntyre '478; see figs. 1 and 3; col. 2, lines 18-30, col. 4, lines 11-66, col. 5, lines 160 and col. 6, lines 1-25).

Re claim 19: McIntyre '478 discloses a method and apparatus for utilizing multiple versions of a page descriptor language comprising:

a central component for receiving the datastream in a first format (i.e. McIntyre '478 discloses a control driver that receives the datastream in a high-level language, or a first format; see figs. 1 and 3; col. 2, lines 18-30, col. 4, lines 11-66, col. 5, lines 160 and col. 6, lines 1-25);

a plurality of sources in the central component, wherein each of the plurality of sources is associated with at least one transform (i.e. the plurality of personalities, considered as sources, are managed by both the control driver (104) and the boot agent

(102). Since the control driver manages the personalities, this can be considered as having the personalities in the control driver (104) to be managed; see figs. 1 and 3; col. 2, lines 18-30, col. 4, lines 11-66, col. 5, lines 160 and col. 6, lines 1-25); and

at least one compute node coupled to the central component (i.e. the printer driver (114), considered as the compute node, is coupled to the control driver (104); see figs. 1 and 3; col. 2, lines 18-30, col. 4, lines 11-66, col. 5, lines 160 and col. 6, lines 1-25),

wherein the central component instantiates at least one source of the plurality of sources (i.e. the control driver (104) uses the printer driver (114) to create a particular realization of a printer description language, or instantiates, through recognizing the personality in the system, which is analogous to the plurality of sources; see figs. 1 and 3; col. 2, lines 18-30, col. 4, lines 11-66, col. 5, lines 160 and col. 6, lines 1-25), and distributes each of the work units to the at least one compute node,

wherein the at least one compute node converts each data work unit into a second format (i.e. in McIntyre '478, a plurality of print jobs, considered as work units can be processed. The printer driver (114), considered as the compute node, processes the incoming print jobs by recognizing the type of PDL is input into the system. The printer driver selects a PDL type from a PDL registry (112) to correspond with the incoming data and this PDL type chosen is used to convert the data into a different format, analogous to the second format, which is a low-level data stream; see figs. 1 and 3; col. 2, lines 18-30, col. 4, lines 11-66, col. 5, lines 160 and col. 6, lines 1-25).

However, McIntyre '478 fails to teach a queue, at least one compute node coupled to the central component via the queue, parses the datastream into a plurality of work units in the first format, wherein each work unit may be processed independent of all other work units, wherein the plurality of work units are parsed from a single job, wherein the central component distributes each of the work units to the at least one compute node by queueing each data work unit on a queue accessible by a plurality of compute nodes and a control work unit to be processed by a computer node, wherein the at least one compute node converts each data work unit into a second format independent of all other compute nodes operable on other work units, and wherein at least two compute nodes are operable in parallel to convert at least two data work units in parallel.

However, this is well known in the art as evidenced by Vennekens '711. Vennekens '711 discloses a queue (i.e. the inventions of McIntyre and Vennekens are both involved with processing PDL for an output in the system (same filed of endeavor). However, in the system, the FIFO queue performs the feature of a queue; see col. 6, lines 19-65); and

at least one compute node coupled to the central component via the queue (i.e. the sub-process (36 or 37) considered as the compute nodes are coupled to the master process (32) considered as the central component through the FIFO queue; see fig. 1, see col. 6, lines 19-65),

parses the datastream into a plurality of work units in the first format (i.e. the apparatus and method of Vennekens '711 divides the PDL data stream to provide a plurality of PDL segments. These segments are in a PDL format, analogous to a first format and the function of parsing is performed by the generation of the PDL data stream into a plurality of independent PDL segments, which the plurality of segments are considered as the plurality of work units; see fig. 1, col. 2, line 55 - col. 3, line 65 and col. 4, lines 24-39),

wherein each work unit may be processed independent of all other work units (i.e. the master process (32) is used to analyze the incoming PDL and generate independent PDL segments that can be processed independently from other segments. The work units are then processed independently in the sub-processes (36 or 37) in the system; see col. 6, lines 19-62),

wherein the plurality of work units are parsed from a single job (i.e. a page that can represent a single job can be described by a plurality of independent PDL data segments; see col. 3, line 66 – col. 4, line 23),

wherein the central component distributes each of the work units to the at least one compute node by queueing each data work unit on a queue accessible by a plurality of compute nodes (i.e. when the PDL data stream is divided into the data or control commands, then the commands are then passed from the master process to a FIFO queue, which is considered as the mechanism that queues work units. The segments are then made available to the sub-processes considered as the compute nodes; see fig. 1, col. 6, lines 19-65), and

a control work unit to be processed by a computer node (i.e. in the system, a PDL data stream segment is assigned a control command and the segment with the control commands is translated with a sub-process (36 or 37), which is considered as a computer node; see col. 2, line 55 - col. 3, line 65),

wherein the at least one computer node converts each data work unit into a second format independent of all other compute nodes operable on other work units (i.e. the master process (32) is used to analyze the incoming PDL and generate independent PDL segments that can be processed independently from other segments. The work units are then processed independently in the sub-processes (36 and 37) in the system; see col. 6, lines 19-62) and

wherein at least two compute nodes are operable in parallel to convert at least two data work units in parallel (i.e. the PDL data stream is divided into segments that are processed in parallel on the sub-processes; see col. 2, line 55 - col. 3, line 65).

Therefore, in view of Vennekens '711, it would have been obvious to one of ordinary skill at the time the invention was made to a queue, at least one compute node coupled to the central component via the queue, parses the datastream into a plurality of work units in the first format, wherein each work unit may be processed independent of all other work units, wherein the plurality of work units are parsed from a single job, wherein the central component distributes each of the work units to the at least one compute node by queuing each data work unit on a queue accessible by a plurality of compute nodes and a control work unit to be processed by a computer node, wherein the at least one compute node converts each data work unit into a second format independent of all other compute nodes operable on other work units, and wherein at least two compute nodes are operable in parallel to convert at least two data work units in parallel in order to generate a plurality of independent PDL data stream segments (as stated in Vennekens '711 col. 2, lines 58-65).

However, the combination of McIntyre ‘478 and Vennekens '711 fails to teach the features of by queuing a scheduled control work unit at a tail of the queue to be processed after all other work units presently in the queue, by queuing an immediate control work unit at a head of the queue to be processed before all other work units in the queue and by forwarding an interrupt control work unit immediately regardless of any work units in the queue.

However, this is well known in the art as evidenced by Salgado ‘621. Salgado ‘621 discloses the feature of by queueing a scheduled control work unit at a tail of the queue to be processed after all other work units presently in the queue (i.e. like the above applied references, the reference of Salgado is used to process PDL information for output (same field of endeavor). However, when looking at EXAMPLE 1 in column 18, this example shows examples of the different types of control work units. The system error report job is considered as the schedule work unit, placed at the end of the queue and not processed until the jobs before the system error report job are processed beforehand; see col. 18, line 40 – col. 20, line 36);

by queueing an immediate control work unit at a head of the queue to be processed before all other work units in the queue (i.e. The net print job is considered analogous to the immediate control work unit since this job is immediately placed at the head of the queue before all work units in the queue. This job is also similar to the copy job listed in the same column 18; see col. 18, line 40 – col. 20, line 36);

by forwarding an interrupt control work unit immediately regardless of any work units in the queue (i.e. the interrupt control work unit is analogous to the Authorized User Job in EXAMPLE 1 in column 19. Once this AU job reaches the system, it is immediately processed and interrupts the job that is currently being processed; see col. 18, line 40 – col. 20, line 36).

Therefore, in view of Salgado ‘621, it would have been obvious to one of ordinary skill at the time the invention was made to have the features of the features of by queuing a scheduled control work unit at a tail of the queue to be processed after all other work units presently in the queue, by queuing an immediate control work unit at a head of the queue to be processed before all other work units in the queue and by forwarding an interrupt control work unit immediately regardless of any work units in the queue incorporated in the device of McIntyre ‘478, as combined with the features of Vennekens '711, in order to have a system with a queue for structuring an order in which a plurality of print jobs is to be processed (as stated in Salgado col. 5, lines 6-29).

Re claim 20: The teachings of McIntyre '478 in view of Vennekens '711 and Salgado ‘621 are disclosed above.

McIntyre ‘478 discloses the system of claim 19, wherein each of the at least one compute nodes loads the at least one transform as a dynamic library (i.e. the printer driver (114), considered as the compute node, is able to utilize a personality, analogous to a transform, in interpreting incoming data. The incoming data is analyzed by the control driver (104) and the link between the incoming data and the appropriate personality to use for interpretation is made. The link of the incoming data to a collection of software used to change or provide services to other programs is an example of a dynamic library; see figs. 1 and 3; col. 2, lines 18-30, col. 4, lines 11-66, col. 5, lines 160 and col. 6, lines 1-25) and utilizes the at least one transforms to convert a work unit in the first format to the second format (i.e. the printer driver (114) utilizes one of the personalities, or transforms, to convert a print job, considered as a work unit, from a high-level language to a low-level language that the printer can understand. This is analogous to converting from a first to a second format; see figs. 1 and 3; col. 2, lines 18-30, col. 4, lines 11-66, col. 5, lines 160 and col. 6, lines 1-25).

Re claim 21: The teachings of McIntyre '478 in view of Vennekens '711 and Salgado ‘621 are disclosed above.

However, McIntyre '478 fails to teach the system of claim 19, wherein the central component further includes: a load balancing mechanism coupled to the at least one source for distributing the plurality of work units to the at least one compute node.

However, this is well known in the art as evidenced by Vennekens '711. Vennekens '711 discloses a load balancing mechanism coupled to the at least one source for distributing the plurality of work units to the at least one compute node (i.e. the inventions of McIntyre and Vennekens are both involved with processing PDL for an output in the system (same filed of endeavor). However, the apparatus and method of Vennekens '711 divides the PDL data stream to provide a plurality of PDL segments.

The function of load balancing among the sub-process is performed; see col. 6, lines 19-62).

Therefore, in view of Vennekens '711, it would have been obvious to one of ordinary skill at the time the invention was made to have a load balancing mechanism coupled to the at least one source for distributing the plurality of work units to the at least one compute node in order to generate a plurality of independent PDL data stream segments (as stated in Vennekens '711 col. 2, lines 58-65).

Re claim 25: The teachings of McIntyre '478 in view of Vennekens '711 and Salgado ‘621 are disclosed above.

McIntyre '478 discloses the system, wherein the at least one source is instantiated as a dynamic library (i.e. when using a transform to convert incoming data into another form, the printer driver (114) requests for a transform to convert from the incoming PDL into a low-level language for the printer to understand. This is performed by linking the incoming data to the specific personality that will perform the transformation of the data stream to the low-level format. This process is similar to a library with a collection of subprograms used to develop other pieces of information used by the system and provide the function of transformation that is linked to a certain input language in the invention of McIntyre '478; see figs. 1 and 3; col. 2, lines 18-30, col. 4, lines 11-66, col. 5, lines 160 and col. 6, lines 1-25).

Conclusion

5. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

6. Stumbo (USP 6084688) discloses a network print server with page-parallel decomposing that discloses distributing jobs to a central controlling part that distributes jobs to decomposers that transforms a job into another form (see figure 1).

7. Birk (USP 5157765) discloses a method and apparatus for pipelined parallel rasterization discloses the feature of having data generated into state independent information that contain their own data processing environments (see column 7).

Any inquiry concerning this communication or earlier communications from the examiner should be directed to CHAD DICKERSON whose telephone number is (571)270-1351. The examiner can normally be reached on Mon. thru Thur. 9:00-6:30 Fri. 9:00-5:00.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Twyler Haskins can be reached on (571)-272-7406. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/C. D./
/Chad Dickerson/
Examiner, Art Unit 2625

/Twyler L. Haskins/
Supervisory Patent Examiner, Art Unit 2625